X-RAY SPECTROMETRY
X-Ray Spectrom. 2001; **30**: 431–434

XRS

Short Communication

Rapid calculation of x-ray absorption near edge structure using parallel computation†

C. Bouldin,[1]* J. Sims,[1] H. Hung,[1] J. J. Rehr[2] and A. L. Ankudinov[2]

[1] National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA
[2] Department of Physics, University of Washington, Seattle, Washington 98195, USA

Received 19 June 2001; Accepted 29 June 2001

Modeling x-ray absorption near edge structure (XANES) requires computationally intensive calculations. We show that parallel processing can reduce the time required for XANES calculations by a factor of up to 50 over standard desktop computers. Parallel processing is implemented in our codes using the Message Passing Interface (MPI) and is portable across most hardware and operating systems. We demonstrate the inverse scaling of the parallel algorithm with the number of processors, and discuss how this approach to parallel processing could be implemented in other multiple-scattering calculations. Faster calculations should improve the applicability of *ab initio* XANES studies to many materials science problems. Published in 2001 by John Wiley & Sons, Ltd.

INTRODUCTION

X-ray absorption spectra (XAS) exhibit energy-dependent modulations of photoelectron scattering intensity, which reflects the local atomic structure and chemical information in a given material. However, extracting this information with precision requires a detailed knowledge of the phase shifts and photoelectron scattering amplitudes in the material.[1] XAS is usually broken into two regions, depending on the strength of photoelectron scattering. The extended x-ray absorption fine structure (EXAFS), above ~50 eV, is determined by single and low-order multiple photoelectron scattering, whereas the x-ray absorption near edge structure (XANES), below ~50 eV, often requires full multiple scattering calculations. Photoelectron scattering was originally determined using empirical standards, i.e. by measurements of EXAFS of materials with known structures to extract the photoelectron scattering functions, and then using those parameters to determine the local atomic structure of materials that had presumptively similar photoelectron scattering factors. In simple cases, this approach was very successful, but it also had serious limitations. The worst of these is that it is almost impossible to use empirical standards in the strong scattering XANES region. This led to sustained, and ultimately successful, efforts to develop *ab initio* computer codes to calculate EXAFS and XANES.[2,3]

The *ab initio* EXAFS calculations are now fast, accurate and easily executed on inexpensive desktop computers, since EXAFS is dominated by relatively weak, low-order multiple-scattering processes. However, XANES calculations remain time consuming, often taking hours or days to complete. Further sophistication in the computer codes, such as relaxation of the muffin-tin approximation and the use of non-spherically symmetric potentials, will improve accuracy but increase computational requirements even further. This computational bottleneck in XANES calculations led us to investigate ways to make the calculations faster. In this paper we demonstrate the use of parallel processing techniques for this purpose. We show how this improvement scales with the number of processors and how our approach may be applied in other calculations that simulate electronic structure and other x-ray processes. For example, the program discussed in this paper, Feff, has recently been modified to calculate x-ray emission spectra including solid-state effects.[4]

PARALLEL PROCESSING IMPLEMENTATION

The simplest type of parallel processing is the task parallelism that is intrinsic in simulating a physical process. For XAS, we calculate the x-ray absorption cross-section as a function of energy. Clearly the calculation at any given energy is independent of the same calculation at other x-ray energies and only a finite number of energies have to be calculated, depending on the resolution of the detector and final state lifetime (typically a few tenths of an electron volt). We therefore exploit this physical parallelism to make simultaneous calculations of XANES, assigning different x-ray energies to each processor, which then make the independent, parallel XANES calculations. The results from the individual processors are then assembled to produce the full XANES spectrum.

The starting point for parallelizing the code is to determine which part of a calculation is the most time consuming. Profiling tests showed that only a small section (about 100 lines that call matrix inversion routines) of the

*Correspondence to: C. Bouldin, National Institute of Standards and Technology, 100 Bureau Drive, Stop 8522, Gaithersburg, Maryland 20899, USA.

DOI: 10.1002/xrs.524

33 000 lines of our f77 FORTRAN code Feff8[3] accounted for about 97% of the total execution time. The specific calculation in this section is an energy loop of large complex matrix inversions required to calculate all multiple scatterings of the photoelectron in the x-ray absorption process. Since a similar matrix inversion is carried out at each x-ray energy, these calculations can be spread across multiple processors in a parallel processing cluster of machines as outlined above. We implement this parallelization scheme using a system called the Message Passing Interface (MPI), which is a standard library for implementing parallel processing.[5] The MPI libraries may be implemented using standard C or FORTRAN, combined with fast communications, usually TCP/IP over 100 Mbit ethernet, in a cluster of typically a few dozen computers. MPI can even be implemented on a single dual- or multiple-CPU machine. The advantages of the MPI are simplicity, portability and the ability to implement parallel processing with minor modifications to an existing program. In our case only a few hundred lines of FORTRAN out of the 33 000 in the original Feff8 program had to be modified or added.

In a parallel processing calculation, a key measure of success is high scalability with the number of processors, i.e. the extent to which the addition of more processors result in faster execution, and how many parallel processors can be added before the improvement saturates. To evaluate the performance of our parallel algorithm, we conducted tests for a sample XANES calculation (full multiple scattering calculations over an energy range of ~100 eV around the K absorption edge of GaN using an 87 atom cluster) on six systems. As representative single-processor systems, we did benchmarks of Feff8 on single processor machines, e.g. (1) a 450 MHz AMD K6-3 running SuSe Linux 6.1, and (2) an Apple PowerMac G4 running at 450 MHz. We then ran the MPI version, FFEFMPI, on four MPI clusters: (3) a cluster of 48 Pentium II 500 MHz systems running Redhat Linux 6.2 connected via 100 megabit ethernet, (4) a similar cluster of Pentium III 400 MHz machines running Windows NT connected by 100 megabit ethernet, (5) a cluster of SGI machines and (6) an IBM PowerPC using up to 16 processors. The results of these calculation speed tests are shown in Fig. 1.

We found that the fastest clusters were about 50 times faster than the single processor Linux system (1). We also found that the processing speed could be predicted, as a function of cluster size, by a simple inverse scaling law $T(N) = \alpha(0.03 + 0.97/N)$, where T is the runtime in seconds, α is a constant that accounts for the speed of a given single processor type and the efficiency of the compiler and N is the number of processors in the cluster. Once the single processor scaling factor α has been is determined for a given cluster, the normalized speed of every cluster scales almost identically. As cluster size is increased, the part of the code that runs in parallel changes from the dominant part of the runtime to a time which tends toward an irreducible minimum. In the limit of large cluster sizes, runtime is then dominated by the ~3% of the original runtime that still executes sequentially, implying that cluster sizes of up to $1/0.03 = 33$ are optimal for our parallel processing algorithm. Although the number

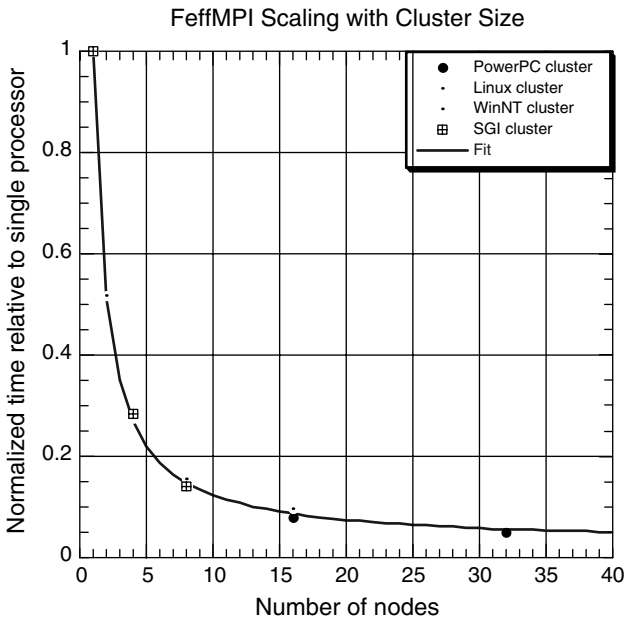

Figure 1. Runtime scaling with the number of processors (nodes) for Linux, Silicon Graphics, Windows NT and IBM PowerPC clusters. Once the calculation times have been rescaled by a factor α to account for processor speed (see text), the scaling is independent of processor and cluster. This scaling can be used to predict the speed up on any cluster, relative to a single processor of that cluster.

33 is system dependent, it is typical of that for XANES calculations.

Task parallel computing using MPI and a homogeneous cluster of computers on a fast ethernet backbone is a generic approach to parallel processing which is adaptable to many scientific calculations, independent of most details of the calculations. Calculations made by scientists simulate physical measurements. The measurements are made as a function of some independent variable (energy, angle, etc.), so the simulation of the measured function is generally amenable to task parallel execution by spreading the values of the independent variable across multiple processors. If the computation spends a fraction β of its total time inside loops over the independent variable, then task parallel execution of the code will reduce the runtime roughly according to $(1 - \beta) + \beta/N$, where N is the number of processors in the cluster. When β is near unity, as in our XANES example where $\beta \approx 0.97$, the computation will benefit greatly from task parallel execution. The only other general requirement is that the time spent communicating between processors must not swamp the time spent in computation; in many cases of practical interest this communication time is very small and can be neglected. To keep this time small, it is important to minimize data transmission, e.g. by confining the matrix inversions in the XANES example within a single processor for a given energy.

Example

To show the utility of our parallel processing approach, we give a brief description of its application to a materials science problem. $BaTiO_3$ (BTO) is one candidate as a high-*k* dielectric

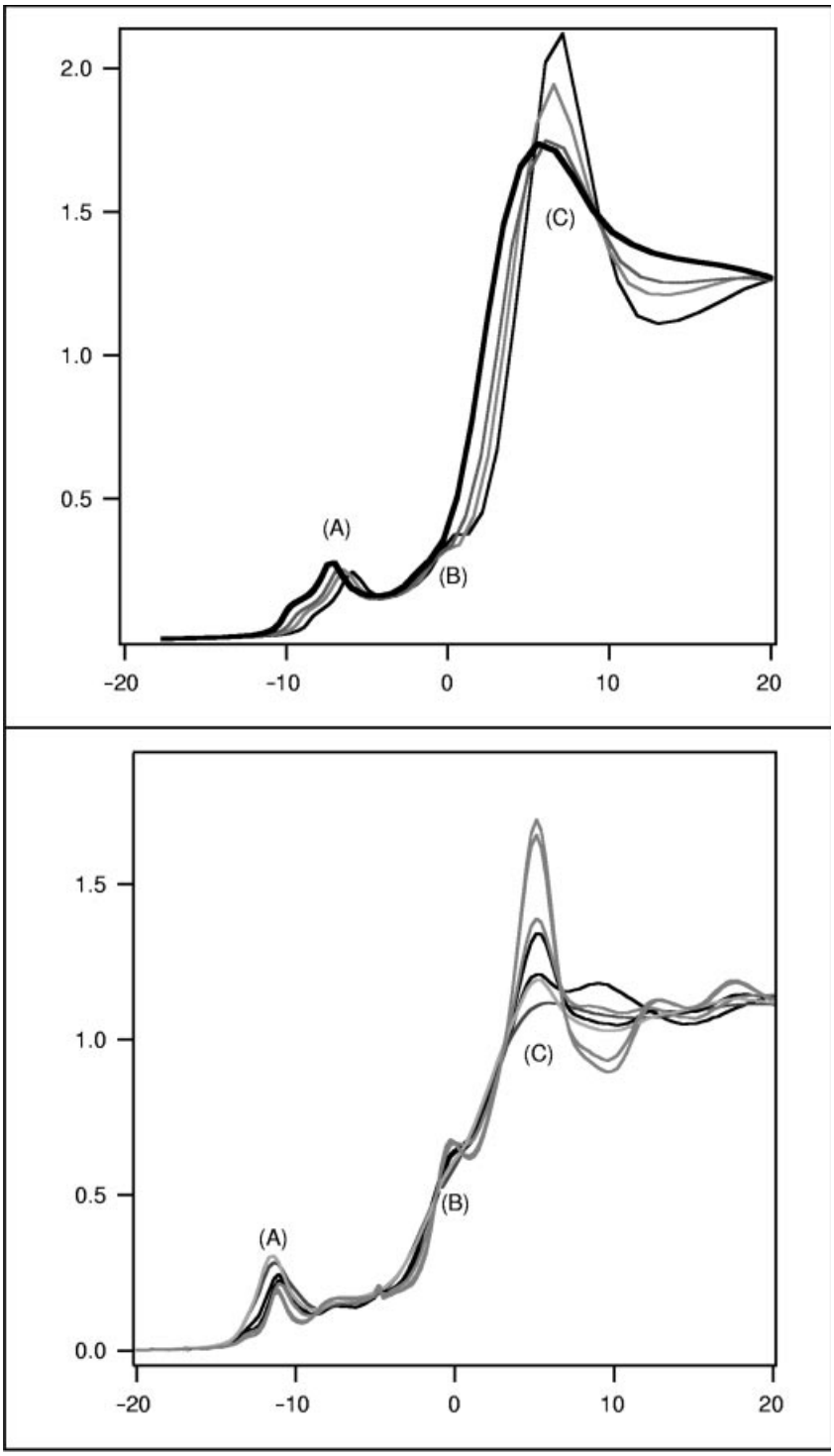

Figure 2. Top panel: series of XANES calculations of the Ti K absorption edge in $BaTiO_3$ as Ba vacancies in the structure increase. The calculation includes atoms within a radius of 4 Å and the calculations show the XANES for $BaTiO_3$ with full (eight atoms) Ba first shell occupancy, then with seven, five and three Ba atoms present. Lower panel: experimental XANES measurements of the Ti K in seven $BaTiO_3$ films deposited on MgO substrates held at various substrate temperatures. The theory reproduces the small energy shift in peak (A), the loss of inflection point (B) and the decrease in peak size (C), suggesting that the $BaTiO_3$ films have large amounts of Ba vacancies in a $BaTiO_3$ structure.

film to replace SiO_2 in DRAM applications,[6] and $SrTiO_3$, which has a very similar structure, is under consideration as a high-*k* gate dielectric in transistor applications.[7] These materials ideally have the perovskite structure, but the applications require ~300 Å thick films deposited at low temperatures, causing the structure to deviate from the bulk crystalline form in unknown ways. For example, in BTO, it is observed that very precise control of the Ba/Ti stoichiometry is required to achieve a high dielectric constant. Deviations of just a few per cent in that ratio can change the dielectric constant by a factor of two, but the atomic level structural changes associated with the change in *k* are unknown.[6]

We measured the XAFS of a series of BTO samples that had large deviations from the ideal BTO stoichiometry but we could not understand the XANES of those samples by using empirical standards, or by Feff calculations based on varying oxygen coordination geometry, e.g. tetrahedral vs octahedral Ti—O bonding. However, using the parallel version of Feff, we modeled our data by holding the BTO structure fixed, while allowing randomly distributed Ba vacancies in amounts consistent with the stoichiometry that we measured by x-ray fluorescence. This gave reasonable agreement with the XANES, reproducing the qualitative features in the XANES, showing the small energy shift of the pre-edge feature (A), loss of inflection point (B) and reduction in the size of peak (C), as shown in Fig. 2. Higher frequencies in the XANES are not reproduced because they are due to higher coordination shells not included in this calculation. Variations in the pre-edge peak are have been associated with varying oxygen coordination geometry, but we observe a smaller pre-edge peak variation in our films that is consistent with Ba vacancies within a structure similar to BTO. Modeling this system relied on the parallel XANES code, even though the only new feature in FeffMPI relative to that described previously[3] is faster calculation speed. Faster calculation speed was essential for this problem, because we achieved turnaround on the calculation in minutes, allowing for many trial calculations in a typical working day. In understanding what was happening in the BTO films, it was critical to reduce the time around the interaction loop of calculating the XANES, viewing the results and comparing with data, tweaking the model and repeating the process. This is a prototype for how parallel computation can speed up the analysis of data by encouraging more user interaction with the modeling program through faster turnaround.

CONCLUSIONS

We have demonstrated a parallel processing approach to XANES calculations using MPI, and found that moderate-sized clusters (~33 processors) typically give a 20-fold speed increase compared with an equivalent single-processor system, and up to a 50-fold improvement compared with typical single-processor desktop systems. The parallel processing algorithm exploits the task or physical parallelism implicit in a XANES calculation and scales well for clusters of up to ~33 processors for the test calculations in this study. The parallel code is nearly as portable as the original sequential code (since MPI libraries are now available for most hardware and operating systems). We have demonstrated portability by running on four different MPI clusters. The task parallel approach used for XAFS calculations can be adapted to other cases of interest in x-ray spectroscopy or electronic structure calculations, since these problems involve similar repeated matrix inversions.

Acknowledgments

We acknowledge the support of Carl Spangler and Hsin Fang in constructing and maintaining the cluster systems used for many of the calculations in this work.

Certain commercial products are identified in the paper for the sake of completeness. This does not constitute an endorsement by the National Institute of Standards and Technology.

REFERENCES

1. Prins R, Koningsberger D (eds). *X-ray Absorption: Principles, Applications, Techniques of EXAFS, SEXAFS and XANES*. Wiley: New York, 1988; 211–253.
2. Rehr JJ, Albers RC. *Rev. Mod. Phys.* 2000; **72**: 621.
3. Ankudinov AL, Ravel B, Rehr JJ, Conradson SD. *Phys. Rev. B* 1998; **58**: 7565.
4. Elam WT, Sieber JR, Ankudinov AL, Rehr JJ. 48th Annual Denver X-ray Conference, *www.dxcicdd.com/01/pdf/F-33.pdf*; *Adv. X-Ray Anal.* 2001; **43**.
5. Gropp W, Lusk E, Skjellum A. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, (2nd edn). MIT Press: Cambridge, MA, 1999; 1–21.
6. Kingon AI, Maria J-P, Streiffer SK. *Nature (London)* 2000; **406**: 1032.
7. Wik GD, Wallace RM, Anthony JM. *J. App. Phys.*, 2001; **89**: 5243.